

1. Q1 For all of these I also plugged in  $x = 1$  million just to confirm my initial beliefs.
  - a. 1 will grow faster. Because  $x$  is being raised to a higher value the  $x^3$  will eventually dominate the equation.
  - b. 1 will grow faster once  $x$  is substantially higher. 1 has the highest polynomial term so that will override all lower polynomials as it goes to infinity. The  $2844x^5$  term will dominate lower values of  $x$  but when I plugged in  $x = 1$  million equation 1 was 6 orders of magnitude greater.
  - c. 2 will grow faster technically. Both of the highest polynomial terms have the same exponent but given that one of them is being multiplied by four that one will grow approximately four times as fast as it approaches infinity, as the second term will drop out.
2. Q2
  - a. Rearranging.  $x = 2^8 = \mathbf{256}$
  - b. Starting off by turning 25 into something with log base 5.  $\log_5(x) = 25$  would be  $5^{25}$  which is about  $2.9 \cdot 10^{17}$ . This turns into  $\log_5(x) = \log_5(2) + \log_5(5^{25})$ . By log addition rules this means  $\log_5(x) = \log_5(2 \cdot 5^{25})$ . Therefore  $x = 2 \cdot 5^{25} = \mathbf{5.9 \cdot 10^{17}}$
  - c. Rearranging.  $4^x = 32$ . Given that  $4^2$  is 16 and  $4^3$  is 64 it is somewhere between those two. Plugging in to my calculator I got  $x = 2.5$ .
3. In this line greetings will contain {"Howdy", "Hello", "Hey"}. In the main statement greetings was made to have hi, hello, and hey as it's strings, but once the change method was called using the string greetings it swapped the first index. This rewrites it, as they are both public and can be changed by other methods.
4. I was having a hard time getting java set up with visual studio code so I just used [Jdoodle](#) to make this instead. I exported it as a java file but if that doesn't work for some reason just paste the following.

```
public class Dice {
    private int numSides;

    public Dice(){
        numSides = 6;
    } // default constructor with a 6 sided die

    public Dice(int sides){
        numSides = sides;
    } // specialized constructor with a n sided die

    public int roll(){
        int out;
        out = (int) (Math.random()*numSides) + 1;
        return out;
    } // roll the die
}
```

```

public static void main(String args[]){ // main method
    Dice dieA;
    dieA = new Dice(20);
    int rollA, rollB, rollC;
    rollA = dieA.roll();
    rollB = dieA.roll();
    rollC = dieA.roll();
    System.out.println(rollA+" "+rollB+" "+rollC);

    System.out.println("Random dnd stat generator, was having fun with java.");
    Dice dieB;
    dieB = new Dice();
    int[] stats = new int[6];
    for(int i = 0; i < stats.length; i++){
        int[] rolls = new int[4];
        rolls[0] = dieB.roll();
        rolls[1] = dieB.roll();
        rolls[2] = dieB.roll();
        rolls[3] = dieB.roll();
        for(int j = 0; j < rolls.length; j++){
            if(rolls[j] <= rolls[0] && rolls[j] <= rolls[1] && rolls[j] <= rolls[2] && rolls[j] <= rolls[3]){
                rolls[j] = 0; //If I set whichever one is lowest to zero immediately then it is set in as
                the lowest for the rest of the loops, then I just add the others together.
            }
        }
        stats[i] = rolls[0]+rolls[1]+rolls[2]+rolls[3];
        System.out.println(stats[i]);
    }
}
}

```